

ЗАДАНИЕ 1 (РАБОТА С ФАЙЛОВЫМ РЕСУРСОМ)

1. Создать в папке `php` файл `form.php`, который будет содержать в себе форму с одним текстовым полем (с именем поля, например, `content`), а также логику обработки этой формы.
2. После заполнения формы и отправки данных на сервер, PHP код должен записать текстовое поле (из формы) в файл `info.txt` при помощи функций `fopen` и `fwrite`. Если файл уже существует, то он должен не перезаписываться, а дополняться, т.е. старые сохранённые текстовые поля остаются, а каждое новое поле записывается с новой строки (символ новой строки - `\n` - чтобы он нормально воспринимался, помещайте его не в одинарные, а в двойные кавычки).
3. Не забываем закрывать файловый ресурс (`fclose`) после работы с ним.
4. Дополнительное условие: если файл `info.txt` уже существует - каждый раз при заходе на страницу открыть ресурс (`fopen`), ссылающийся на него, а затем выбирать построчно все данные (при помощи функции `fgets`) и отображать в виде html таблицы (1 строка - 1 ряд таблицы). Проверку на существование файла `info.txt` (если обратиться к нему для чтения, а его не окажется - будет ошибка) надо осуществлять при помощи функции `file_exists`:

```
<?php

$lines = [];
if (file_exists('info.txt')) {
    // какая-то логика
}
```

Пример таблицы

```
<table>
  <tr>
    <td>1-я колонка 1-го ряда</td>
  </tr>
  <tr>
    <td>1-я колонка 2-го ряда</td>
  </tr>
</table>
```

Пример формы

```
<form action="form.php" method="POST">
  Content:
  <br>
  <textarea name="content"></textarea>
  <br><br>
  <button type="submit" name="submit-form" value="1">
    Submit
  </button>
</form>
```

ЗАДАНИЕ 2 (ОБХОД ДИРЕКТОРИИ)

1. Создайте в папке `php` файл `dir.php`, который будет отображать структуру какой-либо директории (всех файлов, вложенных директорий, файлов вложенных директорий и т.д.).
2. Для отображения создайте функцию `parseDir`, которая принимает строку – путь к необходимой директории (подставьте какую-либо директорию – не очень глубокую, которая находится на вашем компьютере).
3. Структура директории должна отображаться в виде html списка (`ul`), кроме того – директории должны быть обрاملены в тег ``, а файлы – в тег `<i>`
4. Для получения содержимого директории используйте функцию `scandir`, для проверки на файл – `is_file`, а для проверки на директорию – `is_dir`
5. Помните о том, что `.` – запрашиваемая директория, а `..` – родительская директория – если их не игнорировать, то ваш код будет выполняться бесконечно!
6. Не стесняйтесь вызывать функцию `parseDir` внутри самой функции `parseDir` – ведь внутри каждой директории могут быть еще директории и так далее – мы должны достигнуть самого дна!

Пример списка со структурой директорий и файлов:

```
<ul>
  <li><i>file-1.txt</i></li>
  <li><i>file-2.txt</i></li>
  <li>
    <strong>directory-1</strong>
    <ul>
      <li><i>file-3.txt</i></li>
      <li><i>file-4.txt</i></li>
    </ul>
  </li>
</ul>
```

ЗАДАНИЕ 3 (ПОДСЧЕТ ПОСЕЩЕНИЙ)

1. Создайте в папке `php` файл `visits.php`, который будет подсчитывать количество посещений сайта.
2. Если при посещении ссылки <http://php/visits.php> у пользователя не обнаружена cookie (`$_COOKIE`) с количеством посещений (например, с именем `visits`) – устанавливаем ему эту cookie (`setcookie`), а затем пишем – “Это ваш первый визит.”.
3. Если же при посещении ссылки <http://php/visits.php> у пользователя найдена cookie (`$_COOKIE`) с количеством посещений – переустанавливаем эту cookie, увеличив на единицу старое значение (опять `setcookie`). Далее выводим строку – “Вы посещаете эту страницу в `n` раз.”, где вместо `n` должно быть вставлено актуальное на тот момент количество посещений.
4. Помните, что установка cookie должна осуществляться до любого отображения html или вывода на печать PHP значений!

ЗАДАНИЕ 4 (ОГРАНИЧЕННЫЙ ДОСТУП)

1. Создайте в папке `php` два файла - `login.php` и `admin.php`
2. При попытке перехода по ссылке <http://php/admin.php> (файл `admin.php`) инициализируйте сессию, а затем проверяйте, было ли записано в сессию значение `logged_in` - если было записано, то выведите на печать надпись "Добро пожаловать, администратор!". Если значение `logged_in` в сессии отсутствует - перенаправляем посетителя на страницу <http://php/login.php>

Пример перенаправления на `login.php`:

```
<?php
// перенаправление
header('Location: /login.php', true, 302);

/* прекращение исполнения программы - гарантия, что больше
ничего не произойдет */
die();
```

3. При попытке перехода по ссылке <http://php/login.php> (файл `login.php`) опять следует проинициализировать сессию и проверить, записано ли в нее значение `logged_in`. Если было записано - перенаправляем посетителя на страницу <http://php/admin.php> (т.е. нам не надо проходить процедуру аутентификации), если же значение не было записано - отображаем форму с двумя полями - `login` (`input type="text"`), а также `password` (`input type="password"`).
4. При заполнении формы и отправке ее данных на сервер методом POST (отправлять данные следует на тот же самый файл `login.php`). Если в присланных данных существует поле `login` и оно равно `admin`, а также если пришло поле `password` и оно равно `secret` - записываем в сессию значение `logged_in`, равное `true`, а затем перенаправляем посетителя на страницу <http://php/admin.php> Если поле полей нет или они не равны вышеупомянутым значениям - выводим на печать надпись - "Неверный логин или пароль!".